

CLAIMS

1 1. (Original) A method for emulating execution of legacy instructions, where said legacy
2 instructions have instruction addresses, comprising:

3 accessing blocks of said legacy instructions, said blocks having block addresses,
4 storing into a translation store translation information for each of the legacy instructions,
5 storing translation indications, for indicating translated blocks, into an indexing table at
6 block numbers determined by said block addresses,

7 for each particular legacy instruction of a translated block having a particular block number
8 in said table,

9 translating the particular legacy instruction into one or more translated
10 instructions for emulating the particular legacy instruction,

11 if the particular legacy instruction is a store instruction, checking the
12 indications in said table for said particular block number to determine

13 if instruction data has been stored for said particular block number,

14 if instruction data has been stored for said particular block number, checking
15 said translation store to determine if instruction data has been
16 modified; and otherwise, if instruction data has not been stored for
17 said particular block number, bypassing said checking.

1 2. (Original) The method of Claim 1 wherein said step of storing translation indications stores
2 indications for only a subset of all the translated blocks.

1 3. (Original) The method of Claim 2 wherein said subset of all the translated blocks is stored in
2 a cache.

1 4. (Original) The method of Claim 1 wherein said step of storing translation indications uses a
2 subset of block address digits whereby block numbers in said table are the same for multiple
3 different blocks.

1 5. (Original) The method of Claim 4 wherein said block address digits are included in a three digit
2 hexadecimal address field and said subset of block address digits is the center digit.

1 6. (Original) The method of Claim 1 wherein said legacy instructions are for a legacy system having
2 a S/390 architecture.

1 7. (Original) The method of Claim 1 wherein said legacy instructions are object code instructions
2 compiled/assembled for a legacy architecture.

1 8. (Original) The method of Claim 1 wherein said legacy instructions include store instructions for
2 modifying instruction code.

1 9. (Original) The method of Claim 1 wherein said translation indications include a state field for
2 each block number indicating whether the block represented by said block number has been
3 modified.

1 10. (Original) The method of Claim 1 wherein,
2 said step of storing translation indications stores indications for only a subset of all the
3 translated blocks and uses a subset of block address digits whereby block numbers
4 in said table are the same for multiple different blocks,
5 said subset of all the translated blocks is stored in a cache,
6 said translation indications include a state field storing a count for each block number
7 indicating whether the block represented by said block number has been modified,
8 said count in a state field is incremented each time a block represented by said block number
9 has been modified in said cache,
10 said count in a state field is decremented each time a block represented by said block number
11 has been removed from said cache,
12 said bypassing said checking step occurs only when said count is zero.

1 11. (Original) A method for dynamic emulation of object code legacy instructions, where the
2 legacy instructions have instruction addresses determined by compilation/assembly of source code
3 and where the legacy instructions include self-modifying store instructions for modifying instruction
4 code, comprising:

5 accessing blocks of said legacy instructions, said blocks having block addresses,
6 storing into a translation store translation information for each of the legacy instructions,
7 storing translation indications, for only a subset of all the translated blocks, into an indexing
8 table at block numbers determined by said block addresses, said storing translation
9 indications,

10 using a subset of block address digits whereby block numbers in said table
11 are the same for multiple different blocks,

12 including a state field storing a count for each block number indicating
13 whether the block represented by said block number has been
14 modified by self-modifying store instructions,

15 for each particular legacy instruction of said subset of all the translated blocks having a
16 particular block number in said table,

17 translating the particular legacy instruction into one or more translated
18 instructions for emulating the particular legacy instruction,

19 storing said translated instructions in a cache,

20 if the particular legacy instruction is a store instruction, checking the
21 indications in said table for said particular block number to determine
22 if instruction data has been stored for said particular block number,

23 if instruction data has been stored for said particular block number, checking
24 said translation store to determine if instruction data has been
25 modified; and otherwise, if instruction data has not been stored for
26 said particular block number, bypassing said checking.

1 12. (Original) The method of Claim 11 wherein said count in a state field is incremented each time
2 a block represented by said block number has been modified in said cache, said count in a state field
3 is decremented each time a block represented by said block number has been removed from said
4 cache, said bypassing said checking step occurs only when said count is zero.

1 13. (Original) The method of Claim 11 wherein said legacy code is compiled/assembled for a
2 native architecture and executes as a guest on a host architecture.

1 14. (Original) The method of Claim 13 wherein the native architecture employs CISC instructions
2 and the host architecture employs RISC instructions.

1 15. (Original) A system for emulating execution of legacy instructions, where said legacy
2 instructions have instruction addresses, comprising:

3 a group access unit for accessing blocks of said legacy instructions, said blocks having block
4 addresses,

5 a translation store for storing translation information for each of the legacy instructions,
6 an index table for storing translation indications, for indicating translated blocks at block
7 numbers determined by said block addresses,

8 for each particular legacy instruction of a translated block having a particular block number
9 in said table,

10 to translate the particular legacy instruction into one or more translated
11 instructions for emulating the particular legacy instruction,

12 if the particular legacy instruction is a store instruction, to check the
13 indications in said table for said particular block number to determine
14 if instruction data has been stored for said particular block number,

15 if instruction data has been stored for said particular block number, to check
16 said translation store to determine if instruction data has been
17 modified; and otherwise, if instruction data has not been stored for
18 said particular block number, to bypass said checking.

1 16. (Original) The system of Claim 15 wherein said index table stores indications for only a subset
2 of all the translated blocks.

1 17. (Original) The system of Claim 16 including a cache and wherein said subset of all the
2 translated blocks is stored in said cache.

1 18. (Original) The system of Claim 15 wherein said index table stores translation indications using
2 a subset of block address digits whereby block numbers in said table are the same for multiple
3 different blocks.

1 19. (Original) The system of Claim 18 wherein said block address digits are included in a three
2 digit hexadecimal address field and said subset of block address digits is the center digit.

1 20. (Original) The system of Claim 15 wherein said legacy instructions are for a legacy system
2 having a S/390 architecture.

1 21. (Original) The system of Claim 15 wherein said legacy instructions are object code instructions
2 compiled/assembled for a legacy architecture.

1 22. (Original) The system of Claim 15 wherein said legacy instructions include store instructions
2 for modifying instruction code.

1 23. (Original) The system of Claim 15 wherein said index table includes a state field for each block
2 number indicating whether the block represented by said block number has been modified.

1 24. (Original) The system of Claim 15 wherein,
2 said index table stores indications for only a subset of all the translated blocks and uses a
3 subset of block address digits whereby block numbers in said table are the same for
4 multiple different blocks,

5 said subset of all the translated blocks,
6 said system includes a cache for storing,
7 said index table includes a state field storing a count for each block number indicating
8 whether the block represented by said block number has been modified,
9 said count in a state field is incremented each time a block represented by said block number
10 has been modified in said cache,
11 said count in a state field is decremented each time a block represented by said block number
12 has been removed from said cache,
13 said bypassing of said checking occurs only when said count is zero.

1 25. (Original) A system for dynamic emulation of object code legacy instructions, where the
2 legacy instructions have instruction addresses determined by compilation/assembly of source code
3 and where the legacy instructions include self-modifying store instructions for modifying instruction
4 code, comprising:

5 a group access unit for accessing blocks of said legacy instructions, said blocks having block
6 addresses,
7 storing into a translation store for storing translation information for each of the legacy
8 instructions,
9 an index table for storing translation indications, for only a subset of all the translated blocks
10 at block numbers determined by said block addresses, said index table storing
11 translation indications,
12 using a subset of block address digits whereby block numbers in said table
13 are the same for multiple different blocks,
14 including a state field storing a count for each block number indicating
15 whether the block represented by said block number has been
16 modified by self-modifying store instructions,
17 a legacy code translator operating for each particular legacy instruction of said subset of all
18 the translated blocks having a particular block number in said table,

19 to translate the particular legacy instruction into one or more translated
20 instructions for emulating the particular legacy instruction,
21 To store said translated instructions in a cache,
22 if the particular legacy instruction is a store instruction, to check the
23 indications in said table for said particular block number to determine
24 if instruction data has been stored for said particular block number,
25 if instruction data has been stored for said particular block number, checking
26 said translation store to determine if instruction data has been
27 modified; and otherwise, if instruction data has not been stored for
28 said particular block number, to bypass said checking.

1 26. (Original) The system of Claim 25 wherein said count in a state field is incremented each time
2 a block represented by said block number has been modified in said cache, said count in a state field
3 is decremented each time a block represented by said block number has been removed from said
4 cache, said bypassing said checking step occurs only when said count is zero.

1 27. (Original) The system of Claim 25 wherein said legacy code is compiled/assembled for a native
2 architecture and executes as a guest on a host architecture.

1 28. (Original) The system of Claim 27 wherein the native architecture employs CISC instructions
2 and the host architecture employs RISC instructions.